Model-Based Mutation Testing with Action Systems
A Story about Eight Killers

Bernhard K. Aichernig
Joint work with Harald Brandl, Elisabeth Jöbstl, Willibald Krenn, Rupert Schlick (A.I.T)

Institute for Software Technology
Graz University of Technology
Austria

FMCO 2010, Nov 30, 2010
Model-Based Mutation Testing

- UML State Charts: hierarchical, parallel states, constraints over quantors etc. (Papyrus)
- Object-Oriented Action Systems provide formal semantics
- Test-objective based on fault models: mutations in model
- Idea: generate test cases to prevent modeled faults in SUT
- Translator joint work with AIT
- TCG also for Hybrid Systems
Action Systems Semantics

Action System:

\[
\begin{align*}
\textbf{var} & \quad v : T := \text{init} \\
\textbf{methods} & \quad M_1; \ldots; M_n \\
\textbf{actions} & \quad A_1 = g_1 \rightarrow v := e_1; \\
& \quad \ldots; \\
& \quad A_m = g_m \rightarrow M_i(e_i); \\
\textbf{do} & \quad A_1 \\
& \quad \quad \square \\
& \quad \quad A_2; A_3 \\
& \quad \quad // \\
& \quad \quad A_m \\
\textbf{od} & \quad : M_i
\end{align*}
\]

Motivation:

- Well-suited for embedded systems modeling (Event-B)
- Action view maps naturally to LTS testing theories
- Solid foundation:
  - precise semantics
  - refinement
- Compositional modeling
- Many extensions available:
  - object-orientation
  - hybrid systems
Car Alarm System: Interface

«system_under_test»
AlarmSystem

+ alarmArmed
[1]

+ acousticAlarm
[1]

+ opticalAlarm
[1]

Lock
Unlock
Close
Open

«environment»
AlarmArmed
[1]
SetOn()
SetOff()

«environment»
AcousticAlarm
[1]
SetOn()
SetOff()

«environment»
OpticalAlarm
[1]
SetOn()
SetOff()
Car Alarm System: Behaviour

AlarmSystem_StateMachine

- **OpenAndUnlocked**
  - Transition: Unlock
  - Actions: Open, Close

- **ClosedAndUnlocked**
  - Transition: Unlock
  - Actions: Lock, Unlock

- **ClosedAndLocked**
  - Transition: Unlock
  - Actions: Unlock, Lock, Close, Open

- **OpenAndLocked**
  - Transition: Unlock
  - Actions: Close, Unlock

- **Armed**
  - Transition: Unlock
  - Actions: Show Armed /entry, Show Unarmed /exit

- **SilentAndOpen**
  - Transition: Open
  - Actions: Unlock, Close

- **Alarm**
  - **FlashAndSound**
    - Transition: 30 / Deactivate Sound
  - **Flash**
    - Transition: 300
  - Action: Activate Alarms /entry, Deactivate Alarms /exit
76 Alarm System Mutations, e.g.
Test Case Generation

Action System Model

\[ IOLTS^S \]

\[ IOLTS^M \]

discriminating test case

for every mutant

Mutants

ioconf \ldots \text{input-output conformance}
Test Case Generation

Action System Model

 Mutants

\[ IOLTS^S \]

for every mutant

\[ IOLTS^M \]

discriminating test case

ioconf ... input-output conformance
Test Case Generation

Action System Model

IOLTS\textsuperscript{S}

for every mutant

IOLTS\textsuperscript{M}

discriminating test case

ioconf ... input-output conformance
Test Case Generation

Action System Model

 Mutants

 for every mutant

 ioconf ... input-output conformance

\[ IOLTS^S \]

\[ IOLTS^M \]

discriminating test case
Tool Chain

UML\textsuperscript{M} \quad \text{UML to OOAS} \quad \text{OOAS}\textsuperscript{M} \quad \text{Argos} \quad \text{AS}\textsuperscript{M} \quad \text{Ulysses} \quad \text{test cases}

UML \quad \text{OOAS} \quad \text{AS}

Figure: Test Case Generation Tool Chain.

AS\textsuperscript{M} \quad \text{explore}^{LTS} \quad \delta \quad \text{det} \quad \text{SP}_{ioco} \quad \text{product graph} \quad \text{extraction + controllability} \quad \text{test cases}

on-the-fly

Figure: The computation steps of Ulysses.
Additional Mapping: OOAS to CADP

- model simplification
- model checking
- scenario-based TCG in TGV
K1: Brute-Force Mutation Killer

- Given a product graph with a set of fail states.
- Unfold into a tree with maximum depth (10), maximum state revisit of 2
- Linear test case for each path to fail state.
- Inconclusive verdict if path is left
- Motivation: short counter-examples are poor killers
K1: Brute-Force Mutation Killer

<table>
<thead>
<tr>
<th></th>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Depth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Gen. TCs [#]</td>
<td>16</td>
<td>210</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duplicates [#]</td>
<td>12</td>
<td>179</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique [#]</td>
<td>3</td>
<td>469</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCG Time [min]</td>
<td>188</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of Generated TCs
K2: Target-Oriented Mutation Killer

- Given a product graph with a set of fail states.
- One path to every fail state in product graph.
- Inconclusive verdict if path is left.
- Maximum depth (14).
- Motivation: Reduce number of test cases.
## K2: Target-Oriented Mutation Killer

<table>
<thead>
<tr>
<th></th>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Depth</td>
<td>10</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. TCs [#]</td>
<td>16</td>
<td>210</td>
<td>302</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duplicates [#]</td>
<td>12</td>
<td>179</td>
<td>174</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique [#]</td>
<td>3</td>
<td>469</td>
<td>110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. Time [min]</td>
<td>188</td>
<td></td>
<td></td>
<td>91</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of Generated TCs
K3: Adaptive Mutation Killer

- Given a product graph with a set of fail states.
- Shortest path to every fail state in product graph.
- Inconclusive only if this fail state cannot be reached.
- Unbounded depth in finite graph.
- Motivation: Reduce number of test cases
# K3: Adaptive Mutation Killer

<table>
<thead>
<tr>
<th></th>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Depth</td>
<td>10</td>
<td>14</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. TCs [#]</td>
<td>16</td>
<td>210</td>
<td>302</td>
<td>504</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duplicates [#]</td>
<td>12</td>
<td>179</td>
<td>174</td>
<td>217</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique [#]</td>
<td>3</td>
<td>469</td>
<td>110</td>
<td>269</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. Time [min]</td>
<td>188</td>
<td>91</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of Generated TCs
K4: Lazy Mutation Killer

- Is an adaptive killer (K3),
- but checks if an existing test case is guaranteed to cover the next fail state
- New test case only if needed, i.e. when fail state not covered.
## K4: Lazy Mutation Killer

<table>
<thead>
<tr>
<th></th>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Depth</td>
<td>10</td>
<td>14</td>
<td>23</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. TCs [#]</td>
<td>16</td>
<td>210</td>
<td>302</td>
<td>504</td>
<td>129</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duplicates [#]</td>
<td>12</td>
<td>179</td>
<td>174</td>
<td>217</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique [#]</td>
<td>3</td>
<td>469</td>
<td>110</td>
<td>269</td>
<td>123</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. Time [min]</td>
<td>188</td>
<td>91</td>
<td>23</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Number of Generated TCs**
K5: Lazy Ignorant Mutation Killer

▶ Is an adaptive killer (K3)
▶ but checks if an existing test case is able to kill the next mutant
▶ Does not care how he kills!
▶ Any fail state reachable will suffice!
▶ New test cases only when mutant survives.
## K5: Lazy Ignorant Mutation Killer

<table>
<thead>
<tr>
<th></th>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Depth</td>
<td>10</td>
<td>14</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. TCs [#]</td>
<td>16</td>
<td>210</td>
<td>302</td>
<td>504</td>
<td>129</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duplicates [#]</td>
<td>12</td>
<td>179</td>
<td>174</td>
<td>217</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unique [#]</td>
<td>3</td>
<td>469</td>
<td>110</td>
<td>269</td>
<td>123</td>
<td>59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. Time [min]</td>
<td>188</td>
<td>91</td>
<td>23</td>
<td>70</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of Generated TCs
K6: Random First Mutation Killer

- Is an adaptive lazy ignorant mutation killer (K5)
- but starts with a long randomly generated test case (length 150)
K6: Random First Mutation Killer

<table>
<thead>
<tr>
<th></th>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Depth</td>
<td>10</td>
<td>14</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>150 (19)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gen. TCs [#]</td>
<td>16</td>
<td>210</td>
<td>302</td>
<td>504</td>
<td>129</td>
<td>63</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Duplicates [#]</td>
<td>12</td>
<td>179</td>
<td>174</td>
<td>217</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Unique [#]</td>
<td>3</td>
<td>469</td>
<td>110</td>
<td>269</td>
<td>123</td>
<td>59</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Gen. Time [min]</td>
<td>188</td>
<td>91</td>
<td>23</td>
<td>70</td>
<td>23</td>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of Generated TCs
K7: Blind Random Killer

- randomly generated linear test cases
- not mutation-based
## K7: Blind Random Killer

<table>
<thead>
<tr>
<th></th>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Depth</td>
<td>10</td>
<td>14</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>150</td>
<td>(19)</td>
<td>150</td>
</tr>
<tr>
<td>Gen. TCs [#]</td>
<td>16</td>
<td>210</td>
<td>302</td>
<td>504</td>
<td>129</td>
<td>63</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>Duplicates [#]</td>
<td>12</td>
<td>179</td>
<td>174</td>
<td>217</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unique [#]</td>
<td>3</td>
<td>469</td>
<td>110</td>
<td>269</td>
<td>123</td>
<td>59</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>Gen. Time [min]</td>
<td>188</td>
<td>91</td>
<td>23</td>
<td>70</td>
<td>23</td>
<td>10</td>
<td>0.25</td>
<td></td>
</tr>
</tbody>
</table>

Number of Generated TCs
K8: Purposeful Killer

- TCG in CADP-TGV
- 9 handcrafted test purposes
- based on the state chart
## K8: Purposeful Killer

<table>
<thead>
<tr>
<th></th>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max. Depth</strong></td>
<td>10</td>
<td>14</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>150 (19)</td>
<td>150</td>
<td>30</td>
</tr>
<tr>
<td><strong>Gen. TCs [#]</strong></td>
<td>16</td>
<td>210</td>
<td>302</td>
<td>504</td>
<td>129</td>
<td>63</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td><strong>Duplicates [#]</strong></td>
<td>12</td>
<td>179</td>
<td>174</td>
<td>217</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Unique [#]</strong></td>
<td>3</td>
<td>469</td>
<td>110</td>
<td>269</td>
<td>123</td>
<td>59</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td><strong>Gen. Time [min]</strong></td>
<td>188</td>
<td>91</td>
<td>23</td>
<td>70</td>
<td>23</td>
<td>10</td>
<td>0.25</td>
<td>-</td>
</tr>
</tbody>
</table>

**Number of Generated TCs**
CAS Implementation with Injected Faults

<table>
<thead>
<tr>
<th></th>
<th>Mutants</th>
<th>Equiv.</th>
<th>Pairwise Equiv.</th>
<th>Different Faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetState</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Close</td>
<td>16</td>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Open</td>
<td>16</td>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Lock</td>
<td>12</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Unlock</td>
<td>20</td>
<td>2</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Constr.</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>72</td>
<td>8</td>
<td>26</td>
<td>38</td>
</tr>
</tbody>
</table>
Killing Efficiency

Surviving Implementation Mutants:

<table>
<thead>
<tr>
<th></th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetState</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Close</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Open</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Lock</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Unlock</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Constr.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>K5</th>
<th>K6</th>
<th>K7</th>
<th>K8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Killing Rate [%]</td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>97</td>
<td>100</td>
<td>97</td>
<td>66</td>
</tr>
</tbody>
</table>
Conclusions

- **Best:** Random First Mutation Killer
  - Combination of random, lazy and breath-first mutation analysis
- **Worst:** Purposeful Mutation Killer
  - scenarios, test purposes
- We need both!
  - Industry wants scenarios & requirements traceability
- By the way: without UML performance increases 50%
- Ongoing: tool improvement, semantic mutations